

## REMARKS

Reconsideration of the above-identified patent application in view of the amendments above and the remarks following is respectfully requested.

Claims 1, 3-9, 11, 12, 14, 16-31, 33-39, 41, 42, 44, 46-59 and 64-66 are in this case. Claims 20-30 and 50-59 were withdrawn by the Examiner from consideration as drawn to a non-elected invention. In the Office Action mailed July 2, 2007, claims 1, 3-9, 11, 12, 14, 16-19, 31, 33-39, 41, 42, 44, 46-49 and 64-66 were rejected under § 103(a). Independent claims 1 and 31 and dependent claims 65 and 66 now have been amended.

The claims before the Examiner are directed toward a network adapter, such as a host channel adapter (HCA), and a method of its use. The network adapter includes a host interface, an outgoing packet generator, an incoming packet processor, a network output port and a network input port. The host interface couples the network interface adapter to a host processor. The network output port transmits, via a network, outgoing request packets to remote responders and outgoing response packets to remote requesters. The network input port receives, via the network, incoming response packets from remote responders and incoming request packets from remote requesters. The incoming packet processor receives and processes both incoming response packets and incoming request packets such as incoming read request packets. The outgoing packet generator generates outgoing request packets as requested by the host processor and also generates outgoing response packets in response to incoming request packets.

The outgoing packet generator includes a gather engine that gathers, from a system memory accessible via the host interface, and via a commonly shared data flow path, both write data for outgoing request packets and read data for outgoing

response packets. Responsive to an incoming read request packet that specifies data to be read from a system memory accessible via the host interface, the incoming packet processor writes a response descriptor to a memory location, in a memory separate from the network adapter, indicating the data to be read from the system memory. The outgoing packet generator then reads the response descriptor from the memory location and, responsive thereto, generates the outgoing response packet.

**§ 103(a) Rejections – Pettey et al. ‘712 in view of Gasbarro et al. ‘004**

In the Office Action mailed July 2, 2007, the Examiner rejected claims 1, 3-9, 11, 12, 14, 16-19, 31, 33-39, 41, 42, 44, 46-49 and 64-66 under § 103(a) as being obvious over Pettey et al., US Patent No. 6,594,712 (henceforth, “Pettey et al. ‘712”) in view of Gasbarro et al., US Patent No. 6,948,004 (henceforth, “Gasbarro et al. ‘004”). The Examiner’s rejection is respectfully traversed.

Pettey et al. ‘712 teach an InfiniBand target channel adapter (TCA) 202 of an InfiniBand I/O unit 108. The overall structure of TCA 202 is illustrated in Figure 3. TCA 202 includes a transaction switch 302, a bus router 306, IB MACs 308 and PCI bus interfaces 312 and 316. Bus router 306 performs InfiniBand transport layer operations. IB MACs 308 are the interfaces between TCA 202 and an InfiniBand fabric 114. PCI bus interfaces 312 and 316 are the interfaces between TCA 302 and a host of TCA 302. Transaction switch 302 directs packets, datagrams and command messages between IB MACs 308, bus router 306 and PCI bus interfaces 312 and 316. (Transaction switch 302 also includes packet memory blocks 304 in support of direct DRMA operations, which is the invention of Pettey et al. ‘712). The packets that TCA 202 can handle include the packets illustrated in Figures 10-13: SEND packets 1000, RDMA write packets 1100, RDMA read request packets 1200 and RDMA read response packets 1300. It follows that the correspondences between components of

the present invention, as recited in claims 1 and 31, and components of prior art TCA 302 are as in the following table:

Present invention	TCA 202
Host interface	PCI bus interfaces 312 and 316
Network output port	IB MAC 308
Network input port	IB MAC 308
Outgoing packet generator	Bus router 306
Incoming packet processor	Bus router 306

Gasbarro et al. '004 teaches a host fabric adapter 120 for a host 20 that includes a main memory 206. Host fabric adapter 120 includes several DMA Micro-Engines 710 whose function is (column 14 lines 1-4)

...to build, send, receive and acknowledge NGIO/InfiniBand™ packets between the host memory 206...and a serial link...

(Gasbarro et al. '004 say in column 13 line 66 that host fabric adapter 120 includes “one or more” Micro-Engines 710, but, as shown below, host fabric adapter 120 necessarily includes at least two Micro-Engines 710.)

One difference between the present invention as recited in independent claims 1 and 31 and the prior art cited by the Examiner relates to the recited gather engine, that gathers both write data for outgoing write request packets and read data for outgoing read response packets via a commonly shared data flow path.

Figure 14 of Pettey et al. '712 is a block diagram of the logical structure of bus router 306. Bus router 306 includes work queue management logic 1412 for processing InfiniBand work queue requests, transmit packet process logic 1414 for creating outgoing packets, receive packet process logic 1416 for processing incoming packets and completion process logic for maintaining InfiniBand completion queues.

Bus router **306** also includes a work queue memory **1402** in support of work queue management logic **1412**, a TxPP scratchpad memory in support of transmit packet process logic **1414** and a RxPP scratchpad memory in support of receive packet process logic **1416**.

Pettey et al. '714 are silent concerning whether bus router **306** includes the equivalent of the gather engine of independent claims 1 and 31. The Examiner identified the interface adapter illustrated in Figure 7 of Gasbarro et al. '004 as including the functionality of the gather engine of independent claims 1 and 31. That this is not the case can be appreciated from Gasbarro et al. '004 column 14 lines 24-28:

Typically the Micro-Engine (ME) 710 that controls the Send Queue (SQ) may be referred to SQ Micro-Engine (ME), and likewise the Micro-Engine (ME) 710 that controls the Receive Queue (RQ) may be referred to as RQ Micro-Engine (ME). (emphasis added)

In other words, host fabric adapter **120** of Gasbarro et al. '004 has at least two Micro-Engines **710**, one for the Send Queue and one for the Receive Queue. This is the same configuration as that of the prior art HCA described in the specification of the above-identified patent application on page 3 lines 22-24:

To handle the dual role of requester and responder, IB HCAs known in the art typically have separate, independent transmit and receive hardware structures.

In the specific case of host fabric adapter **120** of Gasbarro et al. '004, one Micro-Engine **710** is the transmit hardware structure and the other Micro-Engine **710** is the receive hardware structure. By contrast, the gather engine of independent claims 1 and 31 is a component of “common hardware resources” (page 4 line 21) that handles “both requester and responder communication flows” (page 4 lines 20-21). Specifically, the general operation of the gather engine is described in page 5 lines 8-26 of the specification as follows:

By the same token, in generating RDMA write and send requests to a remote responder, as in preparing RDMA read responses to send to a remote requester, the HCA “gathers” data from the local memory and sends it in packets to a remote destination. Client processes on the local host generate write and send requests by submitting WRs to the HCA, so that WQEs are placed in the appropriate HCA queues. A gather engine services the WQEs by reading the specified data from the local memory and inserting the data in request packets for transmission. To conform to this model, when the HCA receives RDMA read requests from a remote requester, it similarly generates a list of quasi-WQEs in local memory, which identify the data to be sent to the requester. These quasi-WQEs differ semantically from the WQEs generated by the local host, but they are handled by the HCA in the same way. The quasi-WQEs are serviced by the same gather engine that is responsible for servicing the write and send requests. (emphasis added)

In the Office Action mailed July 2, 2007, the Examiner misinterpreted the “common data flow path” of claims 1 and 31 as a “commonly known or used data path”. Therefore, claims 1 and 31 have been amended to clarify that what was intended by the term “common data flow path” was a “commonly shared data flow path”. Support for these amendments is found in the specification in Figure 2, relative to the preferred embodiment of a gather engine, send data engine (SDE) 66, as described on page 20 lines 12-15:

A send data engine (SDE) 66 gathers the data to be sent from the locations in memory 38 specified by the WQEs, via TPT 58, and places the data in output packets for transmission over network 26.

and on page 22 line 30 through page 23 line7:

The execution engine parses each WQE and prepares instructions to SDE 66 regarding a request packet or packets to be sent out. (Similarly, for each quasi-WQE, the execution engine prepares instructions to the SDE regarding the required response packet.) For write and send requests, the SDE gathers the data from memory 38 indicated by the instructions from the execution engine, loads the data into the packets, and passes the packets to output port 68 for transmission.

Specifically, the commonly shared data path is the arrow in Figure 2 from execution unit 60 to SDE 66.

According to MPEP 2143.03,

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

At least the limitation of a gather engine that gathers write data and read data from the system memory via a commonly shared data flow path is neither taught nor suggested by the prior art cited by the Examiner. It follows that independent claims 1 and 31 as now amended are allowable over the prior art cited by the Examiner. It further follows that claims 3-9, 11, 12, 14, 16-19, 33-39, 41, 42, 44, 46-49 and 64-66 that depend therefrom also are allowable.

Another difference between the present invention and the prior art cited by the Examiner relates to the nature of the incoming read request packet and the nature of the response descriptor consequently written by the incoming packet processor. In the Office Action mailed July 2, 2007, the Examiner proposed that the incoming read request packet recited in claims 1 and 31 reads on SEND packet **1000** of Pettey et al. '712, and implied that the consequent writing of a response descriptor by the incoming packet processor is obvious in light of the creation of DRDMA Write WQE **800** by CPU **208** in response to the receipt of SEND packet **1000** (Pettey et al. '712 column 17 line 21) and the statement in column 25 lines 23-26 of Pettey et al. '712 that

...various of the functions performed by the local CPU are capable of being integrated into the TCA, rather than being performed by a processor external to the TCA.

Therefore, to further distinguish the present invention from the cited prior art, dependent claims 65 and 66 have been amended to state that the incoming read request packet is a RDMA read request packet. Support for these amendments is found in the specification, *inter alia* on page 19 lines 27-31:

Similarly, when TCU 52 receives a RDMA read request from a remote requester on a certain QP, it prepares a “quasi-WQE” indicating the required RDMA read response and places the quasi-WQE in a list belonging to the QP in RDB 40. (emphasis added)

In rejecting claims 65 and 66, the Examiner identified the quasi-WQE recited therein with the WQEs illustrated in Pettey et al. ‘712 Figure 7b. That this is a misidentification can be appreciated from the description of InfiniBand Send Queue operations in Gasbarro et al. ‘004 column 7 lines 41-50:

There may be several classes of Send Queue operations, including Send, Remote Memory Access (RDMA), and Memory Binding. For a Send operation, the WQE specifies a block of data in the consumer's memory space for the hardware to send to the destination, letting a receive WQE already queued at the destination specify where to place that data. For an RDMA operation, the WQE also specifies the address in the remote consumer's memory. Thus an RDMA operation does not need to involve the receive work queue of the destination. (emphasis added)

In other words, the InfiniBand standard does not define the creation of a WQE in response to the receipt of a RDMA read request. Note that DRDMA Write WQE **800** of Pettey et al. ‘712 is created in response to the receipt of a SEND packet, not in response to the receipt of a RDMA read request packet. Note, also, that Pettey et al. ‘712 themselves teach against creating a WQE in response to the receipt of a RDMA read request packet, in column 14 lines 61-65:

If the received packet is a RDMA Read Request packet **1200** then no data is transferred by the RxPP logic **1416**. Instead, the RxPP logic **1416** forwards the received packet to the TxPP logic **1414** for creation of an outgoing RDMA Read Response packet **1300**. (emphasis added)

thereby skipping the step of RxPP logic **1416** making (column 14 lines 57-59)

...the association between an incoming IB packet and the appropriate one of the many TCA **202** QPs **712**

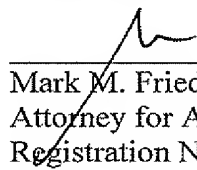
(where the WQEs are according to Figure 7b). The present invention, as recited in dependent claims 65 and 66, remedies this lacuna, to enable the use of the same

gather engine, as described above, to gather data for both requester and responder outgoing packets via a commonly shared data flow path, by creating “quasi-WQEs” that resemble standard InfiniBand WQEs in response to incoming RDMA read request packets.

Finally, some cosmetic amendments (deletion of superfluous “and”s; substitution of semicolons for commas) have been made to claims 1 and 31.

In view of the above amendments and remarks it is respectfully submitted that independent claims 1 and 31, and hence dependent claims 3-9, 11, 12, 14, 16-19, 33-39, 41, 42, 44, 46-49 and 64-66 are in condition for allowance. Prompt notice of allowance is respectfully and earnestly solicited.

Respectfully submitted,



---

Mark M. Friedman  
Attorney for Applicant  
Registration No. 33,883

Date: November 19, 2007